

DOCUMENT RESUME

ED 398 914

IR 055 972

AUTHOR Gregory, Gwen; Brown, M. Marlo
TITLE A Structured Approach to Homepage Design.
PUB DATE 95
NOTE 10p.; In: The Internet--Flames, Firewalls and the Future. Proceedings for the 1995 Conference of the Council for Higher Education Computing Services (CHECS) (Roswell, New Mexico, November 8-10, 1995).
PUB TYPE Speeches/Conference Papers (150) -- Guides - Non-Classroom Use (055)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Graphics; Computer Mediated Communication; Computer Software; *Design Preferences; *Guidelines; Hypermedia; Information Processing; Internet; *Screen Design (Computers); *World Wide Web
IDENTIFIERS Beginning Competence; *Home Pages; HTML; *Structured Programming

ABSTRACT

With no standards governing their creation, a variety of formats are being used for World Wide Web homepages. Some are well organized, present their information clearly, and work with multiple browsers. Others, however, are slow to load, function poorly with some Web browsing software, and are so badly structured that they are very difficult to read. This paper's structured programming approach offers general guidelines for creating Web sites, with an emphasis both on clarity of presentation and on taking advantage of the ease and flexibility of HTML (hypertext markup language) source code. "Structured" HTML is discussed, along with overall suggestions and general guidelines for home page design and construction. Other topics include the planning process for homepages, an overview of Web browsers, and use of graphics. (Author/SWC)

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

A STRUCTURED APPROACH TO HOMEPAGE DESIGN

by: Gwen Gregory,
NMSU
gggregory@lib.nmsu.edu

M. Marlo Brown,
New Mexico State
University
marlo@lib.nmsu.edu

ABSTRACT

The World Wide Web is one of the Internet's greatest areas of growth. People who hardly used the Internet a year ago are now making their own homepages. With no standards governing their creation, a variety of formats are being used for homepages. Some are well organized, present their information clearly and work with multiple browsers. Others, however, are slow to load, function poorly with some Web browsing software and are so badly structured that they are very difficult to read.

This paper is intended for beginning homepage authors and for experienced HTML writers who are interested in new ways to organize information. The authors use a structured programming approach to offer general guidelines for creating WWW sites, with an emphasis both on the clear presentation of information and on HTML source code which is easy understand and allows for future changes.

The paper contains a discussion of "structured" HTML, as well as overall suggestions for homepage design. A list of general guidelines for homepage construction is included. Other topics covered include the planning process for homepages, an overview of Web browsers and using graphics in a homepage.

INTRODUCTION

The concept of structured programming offers a model which can readily be applied to homepage design. The model offers a consistent method of Web page organization which can be used in almost any type of Web site to make the homepage more functional and easier to use. It also makes the homepage easier to update or modify, especially if the changes are made by someone who was not the creator of the homepage.

STRUCTURED PROGRAMMING

Structured programming is not a new concept at all. Popularized in the 1970's, it was used to impose a structure on unstructured programming languages such as FORTRAN, COBOL and BASIC. In his 1978 A Structured Approach to General BASIC, George Ledin identified some general guidelines for structured programming:

- 1) Understand the problem; then break it down into smaller (and possibly simpler) subproblems.
- 2) Choose program structures that can be applied to each subproblem.
- 3) Plan your program's information entry (input) and display (output).
- 4) Do all the necessary bookkeeping (DIMension your array variables, supply the program with explanatory REMarks, and do not neglect the END statement.)

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY

William L. Adkins

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

Ledin's guidelines can be paraphrased for HTML writers in the following manner:

- 1) Evaluate the information being presented and divide it into smaller pieces (pages). Follow "natural" divisions wherever possible.
- 2) Select HTML commands to use with each piece of information.
- 3) Decide on the homepage's input (clickable text, images, forms, etc.) and output (how information will be displayed).
- 4) Perform housekeeping (check with multiple browsers for appearance/functionality, add descriptive, non-displaying comments within the source code, check information for completeness and correctness, test all links).

Structured programming was designed for data processing, not for hypertext. For this reason, this model makes use of structured programming's organizational and documentation concepts but violates one of its cardinal rules: the ban on branching. Ideally, structured programs and their modular subroutines follow a linear model. Each module of code has one entry point and one exit point. This makes sense when programming in BASIC or FORTRAN, as the careless use of GOTO and other branching commands can lead to problems in debugging and updating code, but it is in opposition to one of the greatest strengths of HTML: the ability to set up links to other documents wherever they are needed.

ORGANIZATIONAL CONSIDERATIONS

One of the basic concepts of structured programming is that a problem is broken down into a series of smaller problems, which are then tackled individually. Separate problems are often partitioned off into subroutines, making the program easier to debug and modify for both the programmer and for anyone who has to work on it at a later date. Flowcharting and other visual approaches are often used to help organize a program, frequently before a single line of code is written.

This modular approach is easily applicable to HTML. Where a program is used to solve a problem, a homepage is used to present a collection of information. Instead of subroutines, links to secondary pages are created, each with its own category of information. A modification of the practice of flowcharting is used to assist the creator in keeping track of the information being presented.

Rather than breaking a problem down into smaller problems, an HTML author looks for ways of dividing the collection of information to be presented. The key is to try to use "natural" divisions in the information, rather than trying to impose an artificial system of organization on the information. For example, a computer science professor is creating a homepage. She decides to break it into the following categories: research, teaching, and service. Under research, she includes some descriptions of her research interests, including the full text of

several of her recent journal articles and a complete bibliography of her work dating back to her dissertation. Under teaching, she lists the two courses she is teaching this semester, along with her schedule, office hours, and full text copies of some old tests. Under service, she lists the department, college and university committees she is currently serving on, the professional organizations she belongs to, information on the conferences she recently attended as well as the complete text of the papers she presented there. These divisions are fairly logical, and someone visiting her homepage would probably find the information they were looking for with little trouble.

Systems for the artificial organization of information have their applications in the real world, but they offer little or no advantage in structuring a homepage. For example, many large academic libraries use the Library of Congress call number system to organize their holdings. The LC system is well suited to organizing large collections of books and other works, but it is difficult to use for anyone who is not familiar with it. The professor in the example above could have put everything in her homepage in order by LC call numbers, but in doing so she would have made the information more difficult to find.

Flowcharting

Flowcharting in the realm of programming often involves laying out the sequence of tasks a program is to carry out, showing inputs, decision points, subroutines, outputs and other processes. Flowcharting can easily be adapted to HTML authoring and is extremely helpful in homepage organization. It is especially useful in creating consistency between the various pages within a homepage.

Single sheets of paper -- either letter- or legal-sized -- can be used to represent each page of the homepage. The process can be as simple or as complex as the designer wishes. At its most basic level, such a flowchart would show links to other pages and the information inputs (forms, links allowing comments, etc.) and outputs (display). Flowcharts can also show the page in detail, with notations for highlighted text, justification, graphics, and other elements which will affect how the homepage appears to the users. Homepage flowcharts can also contain notes on the locations of files, the URLs of the links and other structural information. Some experienced homepage designers lay out the entire homepage on paper before beginning to work on the HTML code. Although it may seem time-consuming, flowcharting can save a great deal of time in the long run by helping the designer decide how the information will be presented.

First Page

A structured program often has a "core" section of code which deals with the problem to be solved in a sequential fashion. Decisions within the program are sent to subroutines. The subroutines are often located at the bottom of the list of code, thus leaving the core code intact and easy to read and modify. The HTML counterpart to this concept is the use of a primary file (usually named *index.html*) for the first page of the homepage. This page is used mainly as a link to the other parts of the homepage. Keeping this page as simple as possible not only makes later modifications to the homepage easier, but it also means that the first page of the homepage does not change in outward appearance when the rest of the homepage is updated. The length of the first page should ideally be one to two "screens." Longer, more detailed pages should be linked to, not placed where everyone is forced to wade through them.

Considerations regarding graphics are discussed in detail later in this paper, but the size and number of graphics should be minimized on the first page of a homepage. Inclusion of large images or large numbers of images can slow down the loading of the first page. Avoid following the lead of those who have reduced their first page to a single, clickable icon or graphic which leads to the real homepage. Such an affectation merely slows access for those who are trying to reach the homepage and is especially annoying to repeat visitors.

The first page of the homepage is also the repository for information which is likely to be needed by the majority of visitors to the homepage. In this model, the following pieces of information are suggested.

- ▶ Name of the person, department or organization represented by the homepage. A surprising number of homepages do not give this information.
- ▶ If the purpose of the homepage is not clear from the title, a brief, appealing text introduction can be added.
- ▶ Links to other parts of the homepage, divided into broad subject categories as described above.
- ▶ "Snail mail" (U.S. Postal Service) address. Optional on personal homepages, this should be seen as absolutely essential for businesses and other organizations.
- ▶ Phone (voice) and fax number(s). As with mailing addresses, this is optional for personal homepages.
- ▶ A clickable e-mail address which allows users of the homepage to ask questions or send comments.
- ▶ Date of last update.

Evaluating Homepage Organization

The only way to really test if a homepage is well organized is to have someone else examine it. Having several other people preview it -- especially people with different interests and educational background -- is a good means of finding out if the page will be usable for the Internet community. Some stumbling blocks to watch out for include:

- ▶ Acronyms, jargon, and other subject specific terms are confusing to those outside a narrow field of interest or from a different region or country.
- ▶ All links should be clearly identified. Many HTML writers make the mistake of marking a link with only a word or two, such as "More" or "Bill's Page."

- Identifying each page within one's homepage with a notation or small icon or logo at the top is optional but useful, as it tells users when a link has taken them to another system. It helps in avoiding complaints or comments which should have gone to the owner of a different homepage.

DOCUMENTATION

Page Titles

Titles for the first and subsequent pages within a homepage can be seen as documentation to help the user. Meaningful page titles are a great asset in a homepage, especially now with the large number of WWW search programs available. The main title of the homepage should be as descriptive as possible. When a Web search results in a list of homepage titles and text excerpts, the titles will be the first indication the searcher has as to the contents of the homepage(s) found by the search.

The name of the department, organization or business is a logical choice for a title, as is the name of the individual in the case of a personal homepage. Another alternative for a name is the purpose of the homepage, such as "Mail Order Books" for the homepage of a mail order bookseller. On the other hand, it is counterproductive to use an unrelated or misleading title, such as an acronym instead of a company or organization name. If the homepage is retrieved as part of a WWW search, a name that does not seem to relate to the subject being sought is likely to be bypassed by the searcher in favor of more promising names.

Documenting Source Code

A common practice in most programming is that of placing comments within the source code. These comments may provide explanation of different sections or modules within the program, notes on the organization of the program, information about the writer(s) of the code, date of last update, etc. Comments are good programming practice, and they are also extremely useful in the HTML environment.

Not frequently used in HTML, comments can be added with the following command:

`<!-- text of comment -->`

Comments cannot overlap or nest, but they can cover more than one line. Some browsers, however, may not handle a multiple line comment, so it is best to use a separate command for every line of comments. The comment command should only be used with text. Although it can be used to hide HTML commands from some browsers, this does not work with Netscape.

Comments within the source code may repeat or augment formal, written documentation. The advantage of these comments is that they can be placed directly adjacent to the code they are meant to explain. Additionally, printed documentation can be misplaced and is unavailable to outside users who may look at a Web site and display the source code to see how something was written. Comments in this context can make the HTML code much easier to understand at first glance. Another consideration is that a homepage may have

to be updated or modified by someone other than the creator. Even the original writer may have difficulties deciphering parts of the source code after some time has passed.

TEXT

Most of the actual information presented on a homepage will be text. There are several easy ways to make this text more readable and understandable. Keep the amount of text on introductory pages to a minimum. Conciseness will make the important parts easier for the user to find. For example, the first page might simply have 3 or 4 choices, each of which leads to a more detailed menu of choices, rather than listing all the items in the menus on one page. A small number of logically arranged items, leading to more detailed menus, will help users find the information they want. A menu format, with a list of headings from which to choose, is preferable to including clickable words in paragraphs of text. Reading through paragraphs of text to find the correct entry takes time and effort and is confusing.

The headings (type size) variations of HTML should be used to make the hierarchy of text on the homepage clear. These heading sizes vary from H1 to H6, with H1 being the largest. The main page heading might be in H2 size, the first level subheadings in H3, the second level subheadings in H4, and any other text in unmodified standard size. Keep in mind that users will see slightly different displays of your pages depending on their browser, but use of headings variations enables the homepage writer to specify changes in type size regardless.

Large text files of various sorts can be made available through a homepage. These could include items such as the annual report of an organization, the full text of articles or research reports, or policy and procedure manuals. The ability to access this information remotely via the Web can be very useful. However, these files should be clearly marked so that users know what they are. On the menu from which the user will access them, the size of each file in K should be listed. It should not be necessary for the user to scan through large text files in order to find links to other files. At the bottom of a large file, it is helpful to add buttons to return to the homepage and/or the previous page, to avoid scrolling back through the entire file if reading it online.

GRAPHICS

The ability to display graphics is one of the most exciting aspects of the Web. Graphics can add visual interest and useful information to a homepage. However, overuse or incautious use of graphics can detract from a page's performance or overall appearance. Use of graphics files which are too large can increase loading time of the page to the point where users get impatient and give up, especially repeat users. Graphics files will look different on different monitors and with different browsers, so it is a good idea to preview the appearance of your page on several to check for variations. Files of electronic clip art, including buttons, lines, and icons, are available for downloading through the Web or various ftp sites. These small files can be used to add visual interest to pages without using excessive disk space or loading time. The GIF graphic format is the only one readable by all current browsing software. If you have images in other formats, such as BMP (Windows bitmap) or JPG, a converter program can be used to change them to the GIF format.

Consistency and standardization of graphic content is important. A logo or other similar standardized feature will help the users know when they leave a certain web site's area and link to others. For example, a university might place a small graphic of the university's official logo at the top of each web page. Other graphics can also be standardized. A set of icons for such functions as Home (return to home page), Comments (leave e-mail comments), and Back (return to previous page in the hierarchy) can be created and used on all pages at a particular site. Standardization of other graphics such as buttons, lines, and bars adds unity to a site's pages. A template can be created for use by HTML writers at a particular web site, specifying which graphic images should be used on their pages. A graphics subdirectory can be created on the server as well, so that only one copy of these standardized items needs to be loaded on the server and all pages can link to it, rather than copying it into each page's individual directory. With this method, if a standard graphic is changed or replaced, it need only be done in one place instead of dozens.

Choice of the type of graphic to use is vital. Photographs, while easy to obtain and scan into computer-readable form, do not display well through some browsers and on some monitors. Photo processing software may help clear up some of the problems, by reducing the total number of colors needed for the image and allowing editing of the actual image. However, line drawings are generally smaller size files, require less time to load when viewing the page, and display better. Line drawing of buildings or people can be used instead of photos. If photos are chosen, they should be cropped to remove extraneous information. They may also be reduced in size.

Large image files can take time to load and view. While it is a good idea to avoid overly large graphics files when possible, they are sometimes desirable or necessary. One way to make large images available but avoid forcing all users to spend the time necessary to load them is to use thumbnail graphics. A much smaller copy of the desired image is made. This copy is displayed on the home page, is made clickable, and linked to a separate page which contains the full larger image. This allows only those who are interested to take the time to view the entire image.

Imagemaps are another type of graphic which can be used in homepage creation. Imagemaps are graphics in which various areas of the image are clickable and link the user to other HTML documents. An imagemap with various clickable buttons can be used instead of a menu list. For instance, a map of the state of New Mexico could be made into an imagemap. The various cities and towns in the state would be clickable, with links leading to information on each. If you clicked on Las Cruces, you would link to a page with more detail about the city. Imagemaps, like other graphics, will not be visible to users accessing the page with a text-based browser, so other provisions need to be made for them to access the same information that the imagemap links to. Some users may be unfamiliar with imagemaps and how they work. The address of the link from each area of the imagemap does not display before it is selected, as it does with text links, and users may not realize that they can click on the various areas of the image. A sentence or two explaining how to use the imagemap will help everyone to make full use of the page.

The following guidelines should be considered when using graphics in homepages:

- Don't overuse graphics.
- Use small, appealing, clear images.
- Check how the images appear with different browsers/monitors.
- Standardize graphics for your site.
- Consider using a logo or other identifying image.
- Create a template for graphical appearance of pages at a site

Browsers

Although Netscape is currently the most popular Web browsing software, web developers must remember that users will be viewing their creations with a wide variety of browsing software. These include Lynx, Mosaic, Cello, and MacWeb. Each browser will display information differently. Pages should be tested using at least a couple of these to be sure they work. Some mistakes in writing HTML code will cause errors in Mosaic but be ignored by Netscape, such as forgetting to add quotation marks (") at the end of a link address in an anchor. Many users still have text-based browsers such as Lynx. These users will not be able to view any of the graphic images on your pages. For this reason, any graphics with important contents should have an alternative text file specified. In HTML, this text file is included with the command displaying the graphic. For example, the command `<IMG SRC="library.gif" ALT=[picture of library]>` would display the image library.gif to a graphical browser or the text [picture of library] to a text browser.

SUMMARY

In conclusion, homepage construction based on structured programming concepts makes it simpler to build a functional homepage. It also assists in organization of the homepage and provides for easier maintenance and modification.

Guidelines for Authoring Structured HTML

- Make a flowchart of the page structure.
- Create a template if desired.
- Select graphics; keep them small and simple.
- Use first page as a gateway with links to the rest of the homepage. First page contains only information that will be useful to most or all users.
- Create subsidiary pages for each category of information presented.

- Assign meaningful titles to homepage and links.
- Test homepage with a variety of users/browsing software/computer hardware.
- Document HTML code with non-displaying comments.

References

Aronson, L. (1994). HTML manual of style. Emeryville, CA: Ziff-Davis.

Graham, I. S. (1995). HTML sourcebook. New York: Wiley.

Ledin, G., Jr. (1978). A structured approach to general BASIC. San Francisco, CA: Boyd & Fraser.

APPENDIX

WWW sites containing graphics, HTML writing guides, HTML converters, HTML editors, and other helpful information:

- WWW and HTML Developer's JumpStation
<http://oneworld.wa.com/htmldev/devpage/dev-page.html>
- W3 and HTML Tools
<http://www.w3.org/hypertext/WWW/Tools/Overview.html>
- WWW Repository
<http://cbl.leeds.ac.uk/nikos/doc/repository.html>